

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION CFSTI
DOCUMENT MANAGEMENT BRANCH 410.11

LIMITATIONS IN REPRODUCTION QUALITY

ACCESSION # AD 605021

- ☒ 1. WE REGRET THAT LEGIBILITY OF THIS DOCUMENT IS IN PART UNSATISFACTORY. REPRODUCTION HAS BEEN MADE FROM BEST AVAILABLE COPY.
- ☐ 2. A PORTION OF THE ORIGINAL DOCUMENT CONTAINS FINE DETAIL WHICH MAY MAKE READING OF PHOTOCOPY DIFFICULT.
- ☐ 3. THE ORIGINAL DOCUMENT CONTAINS COLOR, BUT DISTRIBUTION COPIES ARE AVAILABLE IN BLACK-AND-WHITE REPRODUCTION ONLY.
- ☐ 4. THE INITIAL DISTRIBUTION COPIES CONTAIN COLOR WHICH WILL BE SHOWN IN BLACK-AND-WHITE WHEN IT IS NECESSARY TO REPRINT
- ☐ 5. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED, DOCUMENT WILL BE AVAILABLE IN MICROFICHE ONLY.
- ☐ 6. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED DOCUMENT WILL NOT BE AVAILABLE.
- ☐ 7. DOCUMENT IS AVAILABLE IN MICROFICHE ONLY.
- ☐ 8. DOCUMENT AVAILABLE ON LOAN FROM CFSTI (TT DOCUMENTS ONLY).
- ☐ 9.

PROCESSOR:

605021

COPY	OF	22-P
HARD COPY		\$. 1.00
MICROFICHE		\$. 0.50

MULTI-DIMENSIONAL MAXIMIZATION, DYNAMIC
PROGRAMMING AND ECONOMIC LOT SIZE

By

Richard Bellman

P-837


3 April 1956

DDC
The RAND Corporation
1700 MAIN ST. SANTA MONICA, CALIFORNIA
DDC IRA E

Summary

It is shown that some problems arising in the determination of economic lot size lead to the analytic problem of determining the maximum of the function $R(x_1, x_2, \dots, x_n) = \sum_{i=1}^n g_i(x_i)$, subject to a number of constraints of the form $\sum_{j=1}^n h_{ij}(x_j) \leq c_i$, $i = 1, 2, 3$.

These problems are reduced to the determination of a sequence of functions via the functional equation approach of the theory of dynamic programming



MULTI-DIMENSIONAL MAXIMIZATION, DYNAMIC PROGRAMMING AND ECONOMIC LOT SIZE

By
Richard Bellman

1. Introduction

What to produce, and in what quantity? We propose to discuss this fundamental question continually facing the entrepreneur, using the techniques of dynamic programming, with particular emphasis upon the problem of "economic lot size".

In order to make the paper as self-contained as possible, we shall begin by formulating some general allocation processes which include the particular one mentioned above, and indicate the application of the techniques of dynamic programming to the analytic and numerical treatment of these problems.

Following this, we shall treat, in turn, economic lot size problems involving constraints upon capital alone, upon capital and capacity, and finally upon capital, capacity and labor.

In the concluding section of the paper we shall discuss similar problems involving mutually exclusive activities.

2. Allocation Processes— One Dimensional.

Before proceeding to general multi-dimensional processes, let us consider the one-dimensional process. We begin by assuming that we have a quantity of money, x , which we divide into n parts, x_1, x_2, \dots, x_n , with the i -th part allocated to

the 1-th of n activities. From this 1-th activity, we obtain a return of $g_1(x_1)$, whose units we may consider also to be dollars. The problem is to determine the allocation policy which maximizes the total return.

Mathematically, this is the problem of maximizing the function

$$(1) \quad R(x_1, x_2, \dots, x_n) = g_1(x_1) + g_2(x_2) + \dots + g_n(x_n),$$

subject to the constraints

$$(a) \quad x_1 + x_2 + \dots + x_n = x,$$

$$(2) \quad (b) \quad x_1 \geq 0.$$

We shall discuss the numerical aspects of this problem below.

3. Allocation Processes—Multi-Dimensional.

Consider now the multi-dimensional version of the foregoing. Let there be k different resources, in quantities x_1, x_2, \dots, x_k , respectively, which are to be utilized to produce n different types of products. Let

$$(1) \quad x_{1j} = \text{the quantity of the 1-th resource utilized to produce the } j\text{-th product,}$$

$$g_j(x_{1j}, x_{2j}, \dots, x_{kj}) = \text{the return obtained from the allocation of the quantities } x_{1j} \text{ to the production of the } j\text{-th product.}$$

The mathematical problem is that of maximizing the total return,

$$(2) \quad R_n(x_{1j}) = \sum_{j=1}^n g_j(x_{1j}, x_{2j}, \dots, x_{kj}),$$

subject to the constraints

$$(3) \quad \begin{aligned} (a) \quad & \sum_{j=1}^n x_{1j} \leq x_1, \quad i = 1, 2, \dots, k, \\ (b) \quad & x_{1j} \geq 0. \end{aligned}$$

4. Discussion.

A first approach to these problems is by way of the method of Lagrange multipliers. When effective, this method resolves the above problem in an ideal fashion. In practice, however, a number of difficulties occur, which we shall discuss briefly.

a. Non-differentiable functions.

The method of Lagrange multipliers requires that partial derivatives of the various return functions be formed. Two types of difficulties arise. In the first place, the functions may be of complicated analytic form, making the variational equations of little value; in the second place, the functions may be known only approximately, making the derivatives of little significance.

b. Corner Maxima.

In most economic processes, there are constraints of various types upon the independent variables, as in the above

maximization processes. Consequently, it frequently happens that several of the variables attain their upper or lower bounds at the maximum. This means that the maximum point cannot, in general, be obtained by setting partial derivatives equal to zero. Rather, one has to employ a systematic search method, setting some variables equal to their bounds, and taking partial derivatives with respect to the others. This is an inefficient and time-consuming method.

c. Integral Requirements.

In a number of processes, the allocations are not continuous, since the quantities may be required to be integral. This means that the continuous variational method is, at best, an approximation. Sometimes, this approximation is adequate, and sometimes not.

We see that any numerical approach to these problems must be able to overcome the difficulties sketched above. However, more is required of a numerical method. In general, in formulating these problems, we are not so much interested in the numerical solution of any particular problem, as we are in the solution of the family of problems obtained by allowing certain parameters to range over a domain of values of interest. In other words, we are interested in a "sensitivity analysis" of the solution.

The purpose of this sensitivity analysis is to determine the structure of efficient allocation policies, which is to say the dependence of these policies upon the parameters defin-

ing the process.

As we shall see below, the theory of dynamic programming furnishes us a method which is not disturbed by the difficulties cited above, and, in a sense, is more efficient when some of them are present. Furthermore, the method automatically yields a sensitivity analysis, and is designed specifically to study the dependence of efficient policies upon the essential parameters.

5. Dynamic Programming Formulation.

Let us now show how problems of this type may be treated by means of the theory of dynamic programming. A detailed account of the theory may be found in [1], and in a forthcoming book, [2].

Although the process of allocation is static, a single-stage process, it can be reinterpreted to be a dynamic multi-stage process. This we do in the following way. In place of making the allocations, x_1, x_2, \dots, x_n simultaneously, let us think of choosing first x_n , then x_{n-1} , and so on down to x_2 , and finally x_1 . Having decided upon this, let us see upon what the choice of x_k is dependent. At each stage of this multi-stage process we have concocted, the two parameters which specify the state of the process are

- (1) a. the number of stages remaining, k ,
- b. the quantity of resources remaining for allocation, x .

It is clear that the maximum return obtained from the

remaining stages will be a function only of these two parameters. Let us then define the following sequence of functions:

- (2) $f_k(x)$ = the return obtained from k remaining stages when a quantity x of resources remain, and an optimal policy is used.

By a policy, we mean any set of admissible choices of x_k, \dots, x_1 . An optimal policy is a policy which yields the maximum return. Note that the maximum return is uniquely defined, but that there may be many optimum policies.

Beginning with the obvious relation,

- (3) $f_1(x) = g_1(x)$,

assuming, as we shall, that all the functions $g_k(x)$ are monotone increasing, we shall derive a recurrence relation connecting $f_k(x)$ with $f_{k-1}(x)$.

Suppose that a quantity x_k is allocated at the first stage of a k -stage process. A return of $g_k(x_k)$ is obtained, and a quantity $x - x_k$ is available for allocation during the remaining $k-1$ stages of the process. It is clear that whatever the initial allocation x_k , the remaining allocations must be made in such a way as to maximize the total return from the $k-1$ remaining stages. This is a particular application of the "principle of optimality", cf. [1].

Hence the total return from a k -stage process due to an initial allocation of x_k is given by the expression

- (4) $R_k = g_k(x_k) + f_{k-1}(x - x_k)$.

Since $f_k(x)$ is the maximum return from a k -stage process, we obtain the relation

$$(5) \quad f_k(x) = \text{Max}_{0 \leq x_k \leq x} [g_k(x_k) + f_{k-1}(x - x_k)]$$

for $k = 2, 3, \dots$.

This relation determines the sequence recurrently, since $f_1(x)$ is known.

6. Discussion.

The solution of the original n -dimensional maximization, subject to the constraints, has been reduced to the solution of a sequence of n one-dimensional problems. Each of these problems can readily be solved computationally on a digital computer. Furthermore, there are simple methods available for the hand computation of the solutions, using only the graphs of the functions involved.

A digital computer yields not only a table of values of each of the functions $f_k(x)$ for x in a prescribed range $0 \leq x \leq X$, but also a graph of the optimal allocations x_k as a function of x in this range.

For a given x and an n -stage process, the set of n optimal allocations is given by the equations

$$(1) \quad \begin{aligned} x_n &= x_n(n), \\ x_{n-1} &= x_{n-1}(x - x_n), \\ x_{n-2} &= x_{n-2}(x - x_n - x_{n-1}), \\ &\vdots \end{aligned}$$

$$x_2 = x_2(x - x_n - x_{n-1} - \dots - x_3),$$

$$x_1 = x - x_n - x_{n-1} - \dots - x_2.$$

Observe that the solution is given in a form which makes a sensitivity analysis immediate.

Let us now turn to the difficulties discussed in section 4, under the headings a, b, and c.

Since the machine finds the maximum of the function by a systematic search method (which can, in some cases, be greatly speeded up, cf [3]), no difficulty arises from the appearance of non-analytic functions such as $\text{Max}(ax+b, 0)$, or $|cx+d|$.

Again, since the machine determines the maximum value by a search method, constraints of various types aid in this search, rather than hinder, since they narrow down the region of interest. Furthermore, the restriction to integer values similarly aids in the search for a maximum by once again narrowing down the domain of admissible values.

We see that precisely the features which made the variational methods of calculus difficult to apply actually aid the computational solution by means of the functional equation.

Let us show, however, that under certain circumstances, the computation of the solution can be greatly simplified. Suppose that we know that the maximum occurs inside the region of variation for each k . In this case, a maximizing x_k is determined by the equation

$$(1) \quad g'_k(x_k) = f'_{k-1}(x - x_k)$$

and

$$(2) \quad f_k(x) = g_k(x_k) + f_{k-1}(x - x_k),$$

for this value of $x_k = x_k(x)$.

Differentiating, we have

$$(3) \quad f'_k(x) = [g'_k(x_k) - f'_{k-1}(x - x_k)] \frac{dx_k}{dx} + f'_{k-1}(x - x_k) \\ = f'_{k-1}(x - x_k).$$

Referring to equations (2) and (3), we see that only the sequence $\{f'_k(x)\}$ need be computed if we wish only to determine optimal policies.

7. Multi-Dimensional Processes.

Turning to the multi-resource process described in §3, let us define similarly the sequence

$$(1) \quad f_N(x_1, x_2, \dots, x_k) = \text{the return from an } N\text{-stage process starting with initial quantities } x_1, x_2, \dots, \text{ of the } k \text{ resources, } N = 1, 2, \dots$$

As above, the N -stage process consists of allocation only to the first N items.

We have

$$(2) \quad f_1(x_1, x_2, \dots, x_k) = g_1(x_1, x_2, \dots, x_k),$$

and for $N \geq 2$,

$$(3) \quad f_N(x_1, x_2, \dots, x_k) = \max_R [g_N(x_{1N}, x_{2N}, \dots, x_{kN}) \\ + f_{N-1}(x_1 - x_{1N}, x_2 - x_{2N}, \dots, x_k - x_{kN})],$$

where R is the region

$$(4) \quad 0 \leq x_{iN} \leq x_i, \quad i = 1, 2, \dots, k.$$

In theory, there is no limit to the dimension of the problems that we can resolve in this way. In practice, we are limited by the capacity of present day digital computers. There are a number of ways of circumventing these difficulties, which we shall not discuss here, since the applications we shall discuss involve at most three resources.

It is important to note, however, that there is never any restriction upon the number of stages, which is to say the number of activities.

8. Economic Lot Size-Capital Constraint.

Having disposed of these preliminaries, let us now turn our attention to the problem of economic lot size. We shall consider the problem in the following form:

We have an amount of capital x and a choice of the production in varying quantities of N different products. We assume initially that there is an unlimited supply of labor and machines for the production of any items we choose, in any quantity we wish.

If we decide to produce a quantity x_1 of the 1^{th} item, we incur the following costs:

- (1) (a) a_1 = unit cost of raw materials required for the 1^{th} item
- (b) b_1 = unit cost of machine production
- (c) c_1 = unit cost of labor involved
- (d) C_1 = a fixed cost, independent of the amount produced, if $x_1 > 0$.

We also assume a selling price of p_1 per unit for the i^{th} item. The total cost of producing a quantity x_1 of the i^{th} item will then be

$$(2) \quad \sum_{i=1}^N g_i(x_i) ,$$

where

$$(3) \quad g_1(x_1) = (a_1 + b_1 + c_1)x_1 + C_1, \quad x_1 > 0, \\ = 0, \quad x_1 = 0,$$

a discontinuous function at $x_1 = 0$.

Our aim will be to maximize the total profit

$$(4) \quad P_N(x_1, x_2, \dots, x_N) = \sum_{i=1}^N p_i x_i ,$$

subject to the constraints

$$(5) \quad (a) \quad \sum_{i=1}^N g_i(x_i) \leq x ,$$

$$(b) \quad x_i \geq 0.$$

Let

$$(6) \quad f_N(x) = \max_{x_1} P_N(x_1, x_2, \dots, x_N).$$

Then

$$(7) \quad f_1(x) = p_1(x - C_1) / (a_1 + b_1 + c_1), \quad x > c_1, \\ = 0, \quad 0 \leq x \leq c_1 ,$$

and

$$(8) \quad f_N(x) = \max_{\substack{x_N \geq 0 \\ g_N(x_N) \leq x}} [p_N x_N + f_{N-1}(x - g_N(x_N))] .$$

Since $g_N(x_N)$ is discontinuous, we can write this

$$(9) \quad r_N(x) = \text{Max} \left[\begin{array}{l} \text{Max}_{x_N > 0} [p_N x_N + r_{N-1}(x - C_N - (a_N + b_N + c_N)x_N)], \\ r_{N-1}(x) \end{array} \right].$$

It is clear that the term C_N keeps x_N from being too close to zero, and it is easy to see that

$$(10) \quad x_N \geq \frac{r_{N-1}(x) - r_{N-1}(x - C_N)}{p_N}.$$

This lower bound simplifies the search problem.

Although this problem may be solved in explicit terms, we shall not give the details here, since we are primarily interested in the method.

9. Stochastic Version.

Were actual problems as simple and straightforward as the above, the lot of a scientist would be a happy one. We have assumed that the return from a production of a quantity x_k of the k^{th} item is $p_k x_k$. Let us now consider the more realistic situation where the return is a stochastic quantity.

Let $g_k(z)$ represent the cumulative function for the demand z for the k^{th} item. The expected return will then be

$$(1) \quad \begin{aligned} & p_k \int_0^{x_k} z \, dG_k(z) + p_k \int_{x_k}^{\infty} x_k \, dG_k(z) \\ &= p_k \int_0^{x_k} z \, dG_k(z) + p_k x_k (1 - G_k(x_k)). \end{aligned}$$

Supposing that we proceed so as to maximize the expected return, the analytic problem facing us is that of maximizing

$$(2) \quad f_N(x_1, x_2, \dots, x_N) = \sum_{k=1}^N \left[p_k \int_0^{x_k} z \, dG_k(z) + p_k x_k (1 - G_k(x_k)) \right],$$

subject to the constraints

$$(3) \quad \begin{aligned} (a) \quad & x_1 \geq 0, \\ (b) \quad & \sum_{i=1}^N g_i(x_i) \leq x. \end{aligned}$$

As above, this leads to the recurrence relation

$$(4) \quad f_N(x) = \text{Max}_{0 \leq g_N(x_N) \leq x} \left[p_N \int_0^{x_N} z \, dG_N(z) + p_N x_N (1 - G_N(x_N)) + f_{N-1}(x - g_N(x_N)) \right],$$

with

$$(5) \quad f_1(x) = p_1 \int_0^{x_1} z \, dG_1(z) + p_1 x_1 (1 - G_1(x_1)).$$

In a similar fashion, we can treat the problem of determining the allocation policies which maximize the probability of achieving at least a return R .

10. Capital-Machine Constraints.

We have assumed in the foregoing that the only restriction upon production was a limited supply of capital. Let us now consider the case in which we have limited productive capacity as well.

$$(4) \quad \begin{aligned} g_1(x_1) &= a_1(x_1) + b_1(x_1) + c_1(x_1) + C_1, \quad x_1 > 0, \\ &= 0, \quad x_1 = 0. \end{aligned}$$

Setting

$$(5) \quad \text{Max}_{x_1} R = f_N(x, y),$$

we obtain the recurrence relation

$$(6) \quad f_N(x, y) = \text{Max}_D [p_N x_N + f_{N-1}(x - m_N(x_N), y - g_N(x_N))] ,$$

where D is the set of values determined by the simultaneous inequalities

$$(7) \quad a. \quad 0 \leq g_N(x_N) \leq x, \quad 0 \leq m_N(x_N) \leq y.$$

The stochastic version is treated in like fashion.

11. Capital-Machine-Labor Constraints.

Let us now add to the above picture, a constraint on the manpower available to man the machines. Set

$$(1) \quad d_1(x_1) = \text{quantity of manpower required to operate the machines producing a quantity } x_1 \text{ of the } 1^{\text{th}} \text{ item,}$$

and let z be the total quantity of manpower available.

In addition to the constraints of (10.3), we now have the restriction

$$(2) \quad \sum_{i=1}^N d_i(x_i) \leq z.$$

The corresponding recurrence relation is

$$(3) \quad f_N(x, y, z) = \text{Max}_D [p_N x_N + f_{N-1}(x - m_N(x_N), y - g_N(x_N), z - d_N(x_N))] .$$

12. Complementary Constraints

It is often the case that all activities cannot be carried on simultaneously, as a consequence of the specialized nature of the equipment and labor required for the various activities.

Consider a very simple case, where we wish to maximize

$$(1) \quad \sum_{i=1}^{10} p_i x_i,$$

subject to the constraints

$$(2) \quad a. \quad x_i \geq 0$$

$$b. \quad \sum_{i=1}^{10} g_i(x_i) \leq c,$$

$$c. \quad x_1 x_2 = 0, \quad x_3 x_4 x_5 = 0, \quad x_6 x_7 = 0, \quad x_9 x_{10} = 0.$$

Define the sequence of functions

$$(3) \quad (a) \quad f_{10}(c) = \text{Max} \sum_{i=1}^{10} p_i x_i,$$

subject to the above constraints,

$$(b) \quad f_8(c) = \text{Max} \sum_{i=1}^8 p_i x_i,$$

subject to $\sum_{i=1}^8 g_i(x_i) \leq c$, and the first three constraints

in (2c),

$$(c) \quad f_7(c) = \text{Max} \sum_{i=1}^7 p_i x_i,$$

subject to $\sum_{i=1}^7 g_i(x_i) \leq c$, and the first three constraints

in (2c),

$$(d) \quad f_5(c) = \text{Max} \sum_{i=1}^5 p_i x_i$$

subject to $\sum_{i=1}^5 g_i(x_i) \leq c$, and the first two constraints, and

finally

$$(e) \quad f_2(c) = \text{Max} \sum_{i=1}^2 p_i x_i,$$

subject to $\sum_{i=1}^2 g_i(x_i) \leq c$, and $x_1 x_2 = 0$.

Clearly

$$(4) \quad f_2(c) = \text{Max} [p_1 \bar{x}_1, p_2 \bar{x}_2],$$

where $g_1(\bar{x}_1) = c$, $g_2(\bar{x}_2) = c$.

Furthermore

$$(5) \quad f_5(c) = \text{Max}_R [p_3 x_3 + p_4 x_4 + p_5 x_5 + f_2(c - g_3(x_3) - g_4(x_4) - g_5(x_5))],$$

where R is the region

- (6) a. $x_3 x_4 x_5 = 0$
 b. $x_3, x_4, x_5 \geq 0$
 c. $g_3(x_3) + g_4(x_4) + g_5(x_5) \leq c$

It is clear that we can reduce (5) to the equation

$$(7) \quad f_5(c) = \text{Max} \left[\begin{array}{l} \text{Max}_{R_3} [p_4 x_4 + p_5 x_5 + f_2(c - g_4(x_4) - g_5(x_5))], \\ \text{Max}_{R_4} [p_3 x_3 + p_5 x_5 + f_2(c - g_3(x_3) - g_5(x_5))], \\ \text{Max}_{R_5} [p_3 x_3 + p_4 x_4 + f_2(c - g_3(x_3) - g_4(x_4))] \end{array} \right],$$

where

$$(8) \quad R_3 : g_4(x_4) + g_5(x_5) \leq c, \quad x_4, x_5 \geq 0,$$

$$R_4 : g_3(x_3) + g_5(x_5) \leq c, \quad x_3, x_5 \geq 0,$$

$$R_5 : g_3(x_3) + g_4(x_4) \leq c, \quad x_3, x_4 \geq 0.$$

Similarly

$$(9) \quad f_7(c) = \text{Max} \left[\begin{array}{l} \text{Max}_{g_7(x_7) \leq c} [p_7 x_7 + f_5(c - g_7(x_7))] , \\ \text{Max}_{g_6(x_6) \leq c} [p_6 x_6 + f_5(c - g_6(x_6))] \end{array} \right],$$

while

$$(10) \quad f_8(c) = \text{Max}_{g_8(x_8) \leq c} [p_8 x_8 + f_7(c - g_8(x_8))] .$$

Finally

$$(11) \quad f_{10}(c) = \text{Max} \left[\begin{array}{l} \text{Max}_{g_{10}(x_{10}) \leq c} [p_{10} x_{10} + f_8(c - g_{10}(x_{10}))], \\ \text{Max}_{g_9(x_9) \leq c} [p_9 x_9 + f_9(c - g_9(x_9))] \end{array} \right]$$

Bibliography

1. R. Bellman, The Theory of Dynamic Programming, Bull. Amer. Math. Soc., vol 60(1954), pp 503-516.
2. _____, Principles of Dynamic Programming, Princeton University Press, (in press).
3. S. Johnson, Optimal Search is Fibonacciian, (unpublished).